Journal of Intelligent Computing and Networking
https://www.ffspub.com/index.php/jicn/index
ISSN 3079-9228 (print)
ISSN 3107-2798 (online)
E-mail: jicn.office@ffspub.com

Article

A Semantic Communication-Assisted Federated Learning Framework for Internet of Vehicles

Man Luo[1,2], Jin Mao[1,2], Ge Xiong[1,2,†], Muhammad Rizwan Anjum[3]

[1] *School of Computer Science and Technology, Beijing Jiaotong University, Beijing 100044, China*
[2] *Engineering Research Center of Network Management Technology for High Speed Railway of Ministry of Education, Beijing Jiaotong University, Beijing 100044, China*
[3] *Department of Electronic Engineering, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan*

[†] *E-mail: xiongge@bjtu.edu.cn*

Received: January 26, 2026 / Revised: March 2, 2026 / Accepted: March 10, 2026 / Published online: March 30, 2026

Abstract: Federated learning (FL) has wide applications in the internet of vehicles (IoV), but it requires vehicles to perform local training and upload complete model parameters, resulting in heavy communication overhead. To address this issue, this paper proposes a semantic communication-assisted FL (SeFL), which reduces communication overhead by uploading semantic features instead of model parameters. Particularly, the vehicle side employs pre-trained encoders to extract and transmit features, while the server aggregates features for centralized task head training. Moreover, SeFL mitigates the negative impact of non-independent and identically distributed (non-IID) data on model performance through the server centralized training, which aggregates multi-source semantic features to reconstruct global data distribution. Besides, SeFL treats delayed semantic features as new samples participating in iterations, effectively avoiding convergence instability caused by outdated parameter aggregation. Experimental results demonstrate that when using ResNet-101 as the backbone network with a per-round aggregation time threshold ($T_{\text{threshold}}$) of 1.0 s, SeFL achieves classification accuracy improvements of about 8.8% and 18.0% compared to traditional FL (TFL) and hierarchical FL (HFL), respectively. SeFL also reduces total communication overhead by about 99.2% and 98.5% compared to TFL and HFL, respectively. It also shows that when the $T_{\text{threshold}}$ is decreased from 1.0 s to 0.3 s, SeFL maintains relatively stable accuracy while TFL accuracy drops by approximately 28.9%.

Keywords: Semantic Communication; Federated Learning; Internet of Vehicles

1 Introduction

1.1 Background

With the development of Internet of Vehicles (IoV), vehicles are increasingly equipped with sensors and communication modules, enabling real-time collection of large-scale data such as road conditions, traffic flow, and environmental information [1, 2]. Models trained on such data can support perception, planning, and prediction, which are essential for IoV applications including safety alerts, route optimization, and autonomous driving [3–7]. To avoid the heavy communication burden of aggregating raw data, federated learning (FL) has been regarded as a promising solution in IoV to enable collaborative model training by allowing vehicles to train locally and periodically upload model updates to a central server for aggregation [8–11].

Although FL is promising for IoV, several challenges remain in practical deployments. First, exchanging complete model parameters incurs relatively high communication overhead, which strains limited wireless resources [12, 13]. In the process of FL involving multiple vehicles, frequent parameter transmissions consume substantial bandwidth, extend the training cycles per round, and may interfere with the normal communication of other IoV services. Second, communication latency significantly affects model convergence [14, 15]. Due to dynamic network topology and changing wireless channel conditions in IoV, model updates may experience transmission delays. Aggregating outdated parameters reduces convergence speed [16], and delayed updates

[†] Corresponding author: Ge Xiong
[*] Academic Editor: Chunxiao Jiang

negatively impact global model performance [17]. Therefore, communication latency affects both convergence speed and training effectiveness. Third, due to differences in geographical location, driving routes, and sensing hardware among vehicles, IoV data are often non-independent and identically distributed (non-IID) [18], which causes divergence in local model weights and degrades both convergence speed and final accuracy [19]. To address the issues of high communication overhead, communication latency, and non-IID data, semantic communication (SC) has attracted renewed interest.

SC is a newly emerging approach to information transmission that focuses on extracting and transmitting the most relevant semantic information in data [20]. Compared with transmitting complete model parameters, it can substantially reduce communication overhead [21]. In addition, interaction based on semantic information provides opportunities to redesign the FL framework and may help mitigate the impact of communication latency and non-IID data on model synchronization. Therefore, integrating SC into IoV FL is expected to collaboratively address the triple challenges of bandwidth limitations, communication latency, and non-IID data [22].

1.2 Related Work

Up to now, some studies have explored the integration of SC into FL in IoV. In [23], the authors proposed the privacy-preserving personalized FL framework, which split the encoder for training and partially aggregated it, effectively updating semantic encoders in autonomous driving networks. In [24], the authors proposed a SC framework for IoV empowered by FL, which jointly optimized semantic selection and resource allocation based on the semantic-aware algorithm, achieving high semantic extraction accuracy. Overall, these studies verified the feasibility of integrating SC with the FL framework and demonstrated the potential for complementarity between the two frameworks.

Nevertheless, the aforementioned studies have not fully addressed the resource limitations and network dynamics in IoV. Bandwidth constraints, communication latency, and non-IID data remain key bottlenecks [25]. Several studies have started to explore SC-FL integration schemes with better adaptability [26, 27]. Particularly, to reduce communication overhead, in [26], the authors proposed mobility-aware split federated transfer learning, where semantic the encoder and decoder are split into four components. Vehicles kept the pre-trained model and the last decoder layer, while the edge cloud trained the fine-tuning layer and a private decoder. With this design, only the parameters of the last decoder layer need to be aggregated, which reduces communication overhead. However, their study did not consider the impact of non-IID data. To mitigate non-IID effects, in [27], the authors proposed semantic-aware vector-quantized variational autoencoders. The framework uses a contrastive language–image pre-training model to estimate the semantic importance of image patches for semantic compression, and employs vector-quantized variational autoencoders to map continuous features into a discrete codebook, so that only codebook indices are transmitted to reduce transmission overhead. In addition, a hypernetwork is trained at the edge server to generate personalized self-attention layer weights for each vehicle, which allows the model to adapt to local data distributions while retaining shared knowledge, improving accuracy under non-IID data. Although the above mentioned studies reduced communication overhead and addressed non-IID data to some extent, they did not take into account the communication latency caused by dynamic topology and resource constraints in IoV. In practice, aggregating outdated parameters due to delays can severely degrade global convergence [17]. Since the three challenges, i.e., bandwidth limitations, network delay, and non-IID data often coexist in real deployments, the lack of systematic joint optimization makes it difficult to guarantee global model performance in IoV. To the best of our knowledge, no existing work has jointly addressed the aforementioned challenges within an IoV-SC-FL integration framework.

1.3 Motivations and Contributions

To fill this gap, a new FL framework that can reduce communication overhead while ensuring model performance and system stability is designed. However, two key challenges arise in designing such a new FL framework. First, there is a trade-off between communication overhead and model precision [28]. That is, how to effectively reduce the amount of transmitted data to accommodate limited vehicular network resources while ensuring the performance of the final model remains a challenge. Second, the dynamic changes in IoV environment pose a severe challenge to the stability of training. Importantly, the non-IID nature of on-board data and the communication delays caused by dynamic topology often interweave with each other. Moreover, the former challenge leads to the deviation of model weights from the global optimum, while the latter results in outdated parameter aggregation. The combined effect of these two factors seriously disrupts the convergence process of the model. To address these challenges, this paper proposes an SC-assisted FL (SeFL) framework for IoV. The main contributions are summarized as follows:

- An SC-assisted FL framework for IoV, termed SeFL, is proposed. This framework divides the global model into vehicle-side pre-trained encoders and server-side task heads. Vehicle-side clients deploy frozen encoders for forward semantic encoding and upload compressed semantic features to the server. The server aggregates semantic features from each vehicle and centrally trains the classification task head. This design replaces model parameter transmission with semantic feature transmission, offloading the classification task head training process from vehicles to the server, reducing communication overhead while eliminating the need for backpropagation computation on vehicles, thereby reducing the computational burden on vehicles.
- SeFL mitigates the negative impact of non-IID data on global model performance by aggregating semantic features from multiple vehicles on the server and reconstructing a training set that covers the global data distribution. Meanwhile, the training paradigm based on semantic features regards the delayed feature data as new training samples to participate in model iteration, avoiding the

convergence fluctuation problem caused by outdated parameter aggregation in traditional FL (TFL) and enhancing the stability of the system in communication delay environments.

- Experiments show that using ResNet-101 neural network (NN) model with a time threshold ($T_{\text{threshold}}$) of 1.0 s, SeFL achieves about 91.8% classification accuracy, improving by about 8.8% and 18.0% over baseline methods TFL and hierarchical FL (HFL), respectively. SeFL reduces communication overhead by approximately 99.2% and 98.5% compared with TFL and HFL. Moreover, when the $T_{\text{threshold}}$ is reduced from 1.0 s to 0.3 s, which shortens the per-round communication window, SeFL maintains stable performance.

1.4 Notations

Bold lowercase letters denote vectors, bold uppercase letters denote matrices, and italic letters denote scalars. An image is represented as $\mathbf{x}_i \in \mathbb{R}^{H\times W\times C}$, where H, W, and C denote the image height, width and number of channels, respectively. $\mathcal{N}(\mu,\sigma^2)$ denotes a normal distribution with mean μ and variance σ^2. $\|\cdot\|_2$ denotes the ℓ_2 norm, ∇_θ denotes the gradient with respect to θ, and $\mathbf{1}(\cdot)$ denotes the indicator function, which takes the value 1 when the condition is true, and 0 otherwise.

1.5 Organization

The remainder of this paper is organized as follows. Section 2 describes the system framework and communication model of the proposed SeFL. Section 3 presents the training algorithm for SeFL. Section 4 evaluates SeFL through comparative experiments. We conclude the paper in Section 5.

2 The Proposed SeFL

To better describe the presented SeFL, this section first introduces the framework and model update of TFL in IoV, and then presents the framework and workflow of SeFL. Moreover, in view of the dynamic topology changes in IoV, a $T_{\text{threshold}}$ is used to characterize the server waiting window, and a communication delay model is constructed by integrating distance, speed, and channel randomness. We also propose three methods for handling outdated data to meet different scenario requirements.

2.1 TFL in IoV

FL is a distributed machine learning paradigm [29] that enables multiple clients to collaboratively update a global model without sharing raw local data. The core idea is that each client independently updates the model based on its local data, then uploads the updated parameters to a central server for aggregation. The server then distributes the aggregated global model parameters back to each client, and this process is repeated until the global model converges.

TFL in IoV follows the aforementioned FL process. As shown in Figure 1, the system architecture comprises three main components, i.e., multiple vehicle-sides (labeled as "Vehicle-side" at the bottom), a central base station (at the top), and bidirectional wireless communication links between them. Each vehicle maintains a complete NN model consisting of a feature extraction encoder (yellow color) and a classification task head (blue color). The server typically performs federated aggregation using a weighted average, where the weight of each client is proportional to its local data size. The aggregated global parameters are then sent back to vehicles for the next round of updates.

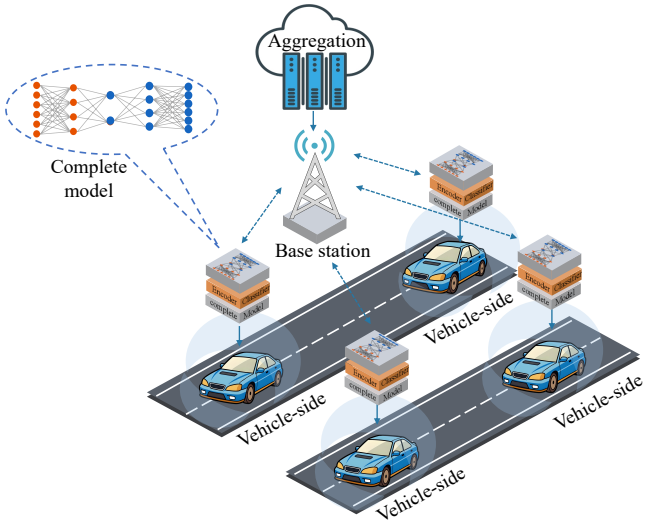


Figure 1: System framework of TFL

In TFL, the vehicle-side acts as a data holder and is responsible for data processing and model updates. This requires the vehicle-side to perform the complete update process using local data and upload the updated model parameters to the server. The server collects model parameters from all vehicles, performs federated aggregation, and redistributes the updated global model to all participating vehicles. The communication network undertakes the two-way transmission task of model parameters between the vehicle-side and the server, and its stability and bandwidth capacity directly affect the update efficiency of the entire FL system.

At the beginning of training, the server initializes the global model $\boldsymbol{\theta}_{\text{global}}^{(0)}$ and broadcasts the model to all participating vehicles through the communication network. Subsequently, each vehicle i receives the global model and performs model updates using its local dataset $\mathscr{D}_i$. The local update process in round t is defined as

$$\boldsymbol{\theta}_i^{(t)} = \boldsymbol{\theta}_{\text{global}}^{(t-1)} - \eta \sum_{j=1}^{|\mathscr{D}_i|} \nabla_{\boldsymbol{\theta}} \mathscr{L}(f(\boldsymbol{\theta}_{\text{global}}^{(t-1)}, \mathbf{x}_j), y_j), \tag{1}$$

where $f(\cdot)$ represents the complete NN training including both forward and backward propagation stages, $\mathscr{L}(\cdot)$ is the loss function, and η is the learning rate. After local updates are completed, each vehicle needs to upload the updated model parameters $\boldsymbol{\theta}_i^{(t)}$ to the server. Upon collecting vehicle parameters, the server performs weighted average aggregation. The parameter aggregation process is defined as

$$\boldsymbol{\theta}_{\text{global}}^{(t)} = \sum_{i=1}^{N} \frac{|\mathscr{D}_i|}{\sum_{k=1}^{N} |\mathscr{D}_k|} \boldsymbol{\theta}_i^{(t)}, \tag{2}$$

where N is the number of vehicles participating in the current round of training, and $|\mathscr{D}_i|$ is the dataset size of vehicle i. Finally, the server distributes the aggregated updated global model to all vehicles, initiating the next round of the cycle.

2.2 Proposed SeFL Framework

Compared with TFL, the proposed SeFL splits the original complete vehicle-side NN model into two separate components, a feature extraction encoder and a classification task head, which are deployed on the vehicle-side and server-side, respectively. As shown in Figure 2, the yellow part on vehicles represents the pre-trained encoders, which extract and upload semantic features along with labels. The blue part on the server represents the classification task head, which downloads the task head model parameters to all vehicles after each training round. Unlike TFL where both encoder and classifier are updated on vehicles, SeFL's vehicle-side encoders remain frozen throughout training and only perform forward inference to generate features. The server-side task head receives features from all vehicles and only broadcasts lightweight classifier parameters back to vehicles. This architecture changes the communication pattern in that vehicles transmit compressed features instead of millions of model parameters, while only receiving the small task head instead of the complete model.

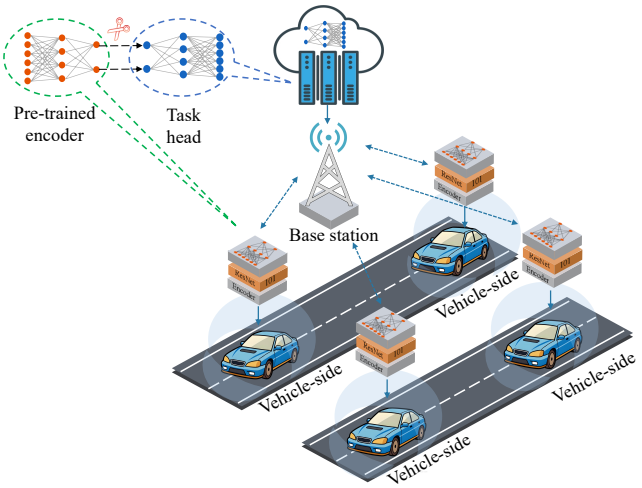


Figure 2: System framework of SeFL

Existing studies have reported that pre-trained visual encoders possess powerful feature representation capabilities and cross-task generalization [30–33]. Therefore, the SeFL proposed in this paper adopts a pre-trained encoder, which is obtained by loading ResNet models pre-trained on the ImageNet dataset. These encoder parameters are frozen before federated training starts and remain fixed throughout the entire training process. On the server-side, the system centrally receives semantic features from the vehicle-side and builds a dataset covering the global data distribution to update the classification task head.

The workflow of the entire system is divided into four stages, including semantic feature extraction, feature transmission, centralized update, and model distribution. Assume that IoV contains V vehicle-sides. The semantic feature extraction process of the i-th vehicle-side is defined as E_{pre}. This process utilizes a pre-trained encoder to achieve the mapping from the original image space to the semantic feature space, i.e., $\mathbb{R}^{H\times W\times C} \to \mathbb{R}^M$. For an input image x_i, the semantic feature extraction process can be expressed as

$$y_i = E_{\text{pre}}(x_i), \tag{3}$$

where $\mathbf{y}_i \in \mathbb{R}^M$ is the extracted M-dimensional semantic feature vector, which preserves the key semantic information of the image while achieving data compression. After the features are uploaded to the server, the system enters the centralized update stage. Specifically, the server merges the received data $\{S_i\}_{i=1}^V$ to construct a dataset covering global data distribution, where $S_i = \{y_j, l_j\}$ is the semantic feature set containing data labels l_j uploaded by the i-th vehicle. The optimization objective on the server-side is to minimize the global loss function $\mathscr{L}_{\text{global}}$, defined as

$$\mathscr{L}_{\text{global}} = \frac{1}{\sum_{i=1}^V |S_i|} \sum_{i=1}^V \sum_{(\mathbf{y}_j, l_j)\in S_i} \ell(C_{\text{global}}(\mathbf{y}_j), l_j) + \lambda \|\boldsymbol{\theta}_{\text{C}}\|_2^2, \tag{4}$$

where $\ell(\cdot,\cdot)$ is the cross-entropy loss function, λ is the regularization coefficient, and the regularization term $\lambda\|\boldsymbol{\theta}_{\text{C}}\|_2^2$ is introduced to constrain the parameter range and prevent model overfitting, where $\boldsymbol{\theta}_{\text{C}}$ denotes the classification task head parameters. For the t-th batch, the update process of the classification task head is expressed as

$$\boldsymbol{\theta}_{\text{C}}^{(t)} = \boldsymbol{\theta}_{\text{C}}^{(t-1)} - \eta \cdot \mathbf{v}^{(t)}, \tag{5}$$

where η denotes the learning rate, and $\mathbf{v}^{(t)}$ represents the accumulated momentum term, i.e., the weighted average of current and historical gradients. After the server completes the centralized update and broadcasts the global classification task head parameters, each vehicle receives them and updates its local classification task head, then proceeds to the next round of iteration until the model converges.

2.3 Communication Model

During the FL process, the server typically waits for vehicle-sides to upload their updates before performing aggregation. In IoV, however, transmission delays vary across vehicles due to dynamic topology and varying channel conditions. Vehicles with large delays may slow down the update process of the global model, and without an effective time limit, the server may end up waiting indefinitely.

To avoid this issue, we introduce a $T_{\text{threshold}}$ on the server, namely a per-round waiting window in which the server only processes updates that arrive on time. To model data timeliness, we further define a maximum delay in rounds, denoted by $S_{\max}$, which limits how many previous rounds a delayed update can remain eligible for aggregation.

The communication model considers factors such as distance, speed, and channel variations [34]. For a data transmission of S bytes, the total transmission time is the sum of data transfer time and network delay, defined as

$$T_{\text{transfer}} = \frac{S}{B_{\text{eff}}}, \tag{6}$$

where the bandwidth is modeled as $B_{\text{eff}} = \frac{B_i \times 10^6/8}{C_i}$ [35], and C_i is the congestion factor introduced to characterize the impact of network congestion on actual throughput. The total transmission time is

$$T_{\text{total}} = T_{\text{transfer}} + T_{\text{delay}}, \tag{7}$$

where T_{delay} is influenced by three components, namely the base delay d_{base}, the velocity factor d_{velocity}, and the random variation factor η. First, to model the effect of transmission distance on delay [36], the base delay d_{base} is defined. As signal attenuation increases with distance, the signal-to-noise ratio decreases, leading to a higher retransmission probability. The base delay is then modeled as a function of communication distance, defined as

$$d_{\text{base}} = \delta_0 + \frac{\sqrt{x_i^2 + y_i^2}}{r} \times \beta, \tag{8}$$

where δ_0 represents the initial delay. The initial vehicle positions are randomly placed within a circular area centered at the base station, with a coverage radius of r meters. (x_i, y_i) denote the vehicle position coordinates, and $\sqrt{x_i^2 + y_i^2}$ represents the distance from the vehicle to the base station, while β is the distance scaling factor. Second, to model the impact of vehicle mobility on communication stability [34], the velocity impact factor d_{velocity} is introduced. High-speed vehicle movement induces Doppler shift, which shortens the channel coherence time and increases the bit error rate. This effect is modeled as

$$d_{\text{velocity}} = \min\left(\frac{S_{v_i}}{S_v}, \varepsilon\right), \tag{9}$$

where S_{v_i} is the i-th vehicle speed in m/s, S_v is the speed limit for urban roads, and ε is the velocity impact limit, determined by the maximum speed limit of urban roads. Third, to model fading effects in real wireless channels, such as multipath propagation and shadowing [37], a normally distributed random perturbation factor $\eta \sim \mathcal{N}(\mu, \sigma^2)$ is introduced. Finally, the transmission delay T_{delay} is given by

$$T_{\text{delay}} = d_{\text{base}} \cdot d_{\text{velocity}} \cdot \eta \cdot \omega, \tag{10}$$

where ω is a scaling factor for global delay. According to the model above, we define the data arrival condition. That is, if the vehicle's uploaded data satisfies $T_{\text{total}} \leq T_{\text{threshold}}$, it is considered on time and is included in the current round of training. Otherwise, the data will be handled in subsequent rounds according to the straggler policy.

Based on straggler handling policies in FL [17, 38], this paper defines three methods to accommodate varying data timeliness requirements. The none policy accepts all data that arrives, regardless of its staleness, and is suitable for scenarios that require maximum data utilization. The drop policy sets the weight of data exceeding the maximum delay rounds to zero, making it suitable for scenarios with high model performance demands. The weight policy uses an exponential decay function to model the relationship between staleness level and data contribution, striking a balance between model performance and data utilization. Defining the staleness rounds as Δr, the delay weighting function can be unified for different straggler handling strategies as

$$w(\Delta r) = \begin{cases} 1.0, & \text{if } policy = \text{'none'}, \\ 1.0, & \text{if } policy = \text{'drop' and } \Delta r \leq S_{\max}, \\ 0.0, & \text{if } policy = \text{'drop' and } \Delta r > S_{\max}, \\ \gamma^{\Delta r}, & \text{if } policy = \text{'weight'}, \end{cases} \quad \gamma \in (0,1), \tag{11}$$

where γ represents a weight parameter. If each vehicle is assigned a weight, then $w_i \triangleq w(\Delta r_i)$. Furthermore, to model packet loss in IoV communication due to factors like channel interference and network congestion, referring to [39], a transmission failure probability model is formulated to be

$$P_{\text{fail}} = P_{\text{rob}} \times (1 - R) \times \tau^{(N)}, \tag{12}$$

where P_{rob} is the base packet loss rate, R is the network reliability parameter, $\tau^{(N)}$ is the retry decay factor [40], and N is the maximum number of retransmissions. This model characterizes the impact of the retransmission mechanism on transmission reliability. After the initial transmission failure, the data enters a retry loop, recalculating the transmission failure probability P_{fail} each time. If all attempts within the maximum retransmission count fail, the data packet is discarded.

3 Training Method of SeFL

To explain the training process of SeFL, this section breaks down the method into three parts, i.e., the overall training process, semantic feature extraction on the vehicle-side, and centralized training on the server-side. Specifically, the overall training process describes the system initialization, the execution flow of the main training loop, and the coordination between components. The semantic feature extraction on the vehicle-side details the process of converting raw images into semantic feature vectors. The centralized training on the server-side presents the method for optimizing the classification task head based on aggregated features.

3.1 Overall Training Scheme of SeFL

The overall training process of SeFL must coordinate interactions between vehicles and the server while accounting for uncertainty due to network fluctuations in IoV environments. To fully utilize data, the algorithm introduces an outdated data processing strategy. The training procedure consists of initialization, iterative model updates, and outputting the final result.

As shown in Algorithm 1, the setup phase consists of server-side and vehicle-side procedures. On the server-side, the classification task head parameters are initialized and the optimizer is configured. On the vehicle-side, each vehicle loads the pre-trained encoder and prepares the transmitted index set. The iterative model update is the core execution stage of SeFL, which sequentially executes four steps of vehicle-side feature extraction, communication delay processing, server-side training, and model broadcasting. During the vehicle-side feature extraction stage, each vehicle invokes

Algorithm 2 to extract semantic features, packages them for transmission, and calculates the corresponding transmission time. If the transmission time exceeds the $T_{\text{threshold}}$ of the current round, the delayed data are assigned appropriate weights according to the weight policy in the delay strategy. Then, in the server-side training stage, the arrived features are sampled to construct a training set, Algorithm 3 is executed to update the model parameters, and the updated parameters are broadcast to all vehicles.

Algorithm 1 SeFL Overall Training Algorithm

Require: Vehicle-side set $V = \{v_1, v_2, \ldots, v_V\}$, encoder E_{pre}, training rounds R, learning rate η
Ensure: Trained classification task head parameters $\boldsymbol{\theta}_C^{(R)}$

Server-side Initialization:
$\boldsymbol{\theta}_C^{(0)} \leftarrow init()$
$\mathscr{O} \leftarrow Optimizer(\boldsymbol{\theta}_C^{(0)}; \eta, \mu, \ldots)$
$feature_{server} \leftarrow \varnothing$
Vehicle-side Initialization:
for each vehicle-side $v_i \in V$ **do**
  $v_i.encoder \leftarrow E_{pre}$
  $v_i.cache \leftarrow \varnothing$
end for
Iterative Model Update:
for $r = 1, \ldots, R$ **do**
  $Arrived[r] \leftarrow \varnothing$
  for each vehicle $v_i \in V$ **do**
    Execute Algorithm 2: Vehicle-side semantic feature extraction
    Compute transmission time: $T_{total} \leftarrow GenerateDelay(v_i)$
    if $T_{total} \leq T_{threshold}$ **then**
      Add features to $Arrived[r]$
    end if
  end for
  $feature_{server} \leftarrow feature_{server} \cup Arrived[r]$
  $\mathscr{T}_r \leftarrow SampleFrom(feature_{server}, batch\_size)$
  Execute Algorithm 3: Server-side centralized training
  $BroadcastModel(\boldsymbol{\theta}_C^{(r)}, V)$
end for
return $\boldsymbol{\theta}_C^{(R)}$

3.2 Vehicle-Side Semantic Feature Extraction

Vehicle-side semantic feature extraction primarily transforms raw image data into semantic features. SeFL uses frozen pre-trained encoders, with vehicles performing only forward inference.

As shown in Algorithm 2, the algorithm consists of three steps, i.e., image preprocessing, semantic feature extraction, and feature normalization. Specifically, in the image preprocessing step, bilinear interpolation is used to resize the input image to a uniform size, followed by standardization. In the semantic feature extraction step, the frozen pre-trained encoder is used to perform forward encoding on the image, generating semantic feature vectors. The output dimensionality varies by encoder. Specifically, ResNet-18/34 produce 512-dimensional features while ResNet-50/101 produce 2048-dimensional features, achieving compression ratios of 293.4:1 and 73.5:1 respectively from the original $224 \times 224 \times 3$ input. In the feature normalization step, L2 normalization is applied to ensure unit norm, computed as $\mathbf{y}_i \leftarrow \mathbf{y}_i / \|\mathbf{y}_i\|_2$, which stabilizes gradient flow during server-side training. Features are then transmitted directly in FP32 format (4 bytes per value) without additional quantization or compression to preserve semantic fidelity.

Algorithm 2 Vehicle-Side Semantic Feature Extraction

Require: Raw data $\mathbf{x}_i$, encoder E_{pre}
Ensure: Semantic feature $\mathbf{y}_i$

Image Preprocessing:
$\mathbf{x}_i \leftarrow Resize(\mathbf{x}_i, H \times W)$
$\mathbf{x}_i \leftarrow Normalize(\mathbf{x}_i, \mu, \sigma)$
Semantic Feature Extraction:
$\mathbf{y}_i \leftarrow E_{\text{pre}}(\mathbf{x}_{i\_\text{processed}})$
Feature Normalization:
$\mathbf{y}_i \leftarrow L2\_\text{Normalize}(\mathbf{y}_i)$
return $\mathbf{y}_i$

3.3 Server-Side Centralized Training

Server-side centralized training is the core of SeFL. By aggregating semantic features from multiple vehicles to build a global dataset, it reduces the impact of non-IID data on model performance.

As shown in Algorithm 3, after the server receives the semantic features from each vehicle, it first initializes the classification task head parameters, momentum, and iteration counter. It then enters the training loop, updating in batches iteratively, computing the global loss, obtaining gradients, and updating parameters. After training, the final parameters are returned.

Algorithm 3 Server-Side Centralized Training

Require: Semantic feature set $\{S_i\}_{i=1}^{V}$, learning rate η, momentum coefficient μ, regularization coefficient λ
Ensure: Updated classification task head parameters $\boldsymbol{\theta}_{\text{C}}$

Initialization:
$\boldsymbol{\theta}_{\text{C}}^{(0)} \leftarrow 0$, $\mathbf{v}^{(0)} \leftarrow 0$, $t \leftarrow 0$
for epoch $= 1$ to max_epochs **do**
  $\mathscr{D} \leftarrow \text{Shuffle}(\bigcup_{i=1}^{V} S_i)$
  for each batch B **do**
    $t \leftarrow t + 1$
    Compute loss: $\mathscr{L}_{\text{global}}$
    Compute gradient: $\mathbf{g}^{(t)} \leftarrow \nabla_{\boldsymbol{\theta}} \mathscr{L}_{\text{global}}(\boldsymbol{\theta}_{\text{C}}^{(t-1)}, B)$
    Update momentum: $\mathbf{v}^{(t)} \leftarrow \mu \cdot \mathbf{v}^{(t-1)} + \mathbf{g}^{(t)}$
    Parameter update: $\boldsymbol{\theta}_{\text{C}}^{(t)} \leftarrow \boldsymbol{\theta}_{\text{C}}^{(t-1)} - \eta \cdot \mathbf{v}^{(t)}$
  end for
end for
return $\boldsymbol{\theta}_{\text{C}}^{(final)}$

4 Experimental Validation

To validate the effectiveness of the proposed SeFL framework, this chapter compares SeFL with three baseline methods. As shown in Figure 3, we construct an IoV simulation scenario to validate the performance of SeFL. Specifically,

within the coverage area of a base station with radius r, V vehicles are randomly distributed [41]. Each vehicle is equipped with a frozen pre-trained encoder E_{pre} for local semantic feature extraction, while the server maintains and trains only the classification task head parameters θ_C. In each SeFL round, vehicle $v_i \in \{1,2,\ldots,V\}$ performs forward encoding on local images to generate feature-label pairs $\{y,l\}$ and uploads them to the server. The server determines whether the received data arrive within the aggregation time window $T_{\text{threshold}}$ and assigns weights to delayed data according to the policy $w(\Delta r)$ before training. Finally, the updated θ_C is broadcast to all vehicles to complete the iterative update. The remainder of this chapter is organized as follows, i.e., Section 4.1 outlines the experimental setup, including the dataset, communication conditions, NN framework, baseline methods, and evaluation metrics. Section 4.2 presents some experimental results to discuss the effectiveness of SeFL, validating SeFL's classification accuracy and stability under various network conditions, with a particular focus on the model's performance as network conditions change. Section 4.3 compares the communication overhead of each method, highlighting SeFL's advantage in reducing communication costs. Section 4.4 assesses the convergence speed of each method by measuring the iteration time per round, demonstrating SeFL's efficiency in enhancing training performance. Section 4.5 further evaluates the robustness of SeFL under different degrees of non-IID data distributions.

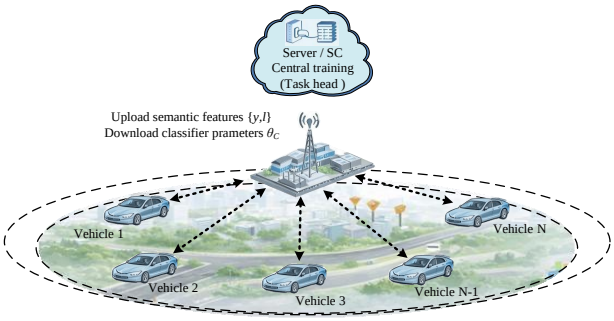


Figure 3: Simulation scenario of SeFL in IoV environment

4.1 Experimental Settings

Dataset: As shown in Figure 4, experiments are conducted using the german traffic sign recognition benchmark (GTSRB) [42], which includes traffic sign images from 43 categories. We use 35,288 images for training and 3,921 images for testing. All images are resized to 224×224 pixels and standardized with mean $\mu = [0.485, 0.456, 0.406]$ and standard deviation $\sigma = [0.229, 0.224, 0.225]$. Data augmentation techniques, such as random horizontal flipping, rotation, and brightness adjustment, are applied to the training data.

Figure 4: Sample Images from GTSRB Dataset

To validate SeFL's advantage in handling non-IID data, the 43 traffic sign categories of the GTSRB dataset are assigned exclusively to vehicles, as shown in Figure 5. Each category is assigned to a single vehicle, with no overlap between categories, ensuring a non-IID data distribution across vehicles. The exclusive category assignment is based on several key considerations. Firstly, in practical deployment, vehicles operating in different geographical regions or at different times naturally encounter different traffic sign distributions. Urban vehicles mainly encounter speed limit and pedestrian crossing signs, while highway vehicles mainly encounter directional signs. Such route-dependent specialization leads to a natural skew in the distribution of scenarios. Secondly, the extreme non-IID setting validates algorithm robustness under the most adverse conditions, ensuring reliable performance across all non-IID levels.

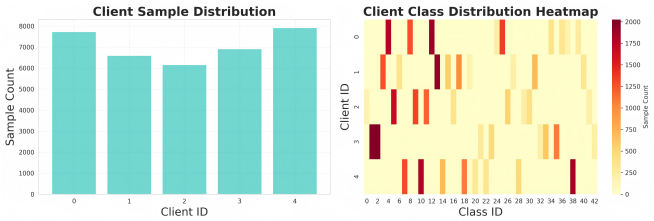


Figure 5: Results of Client Data Partitioning

After category assignment, each vehicle-side further splits its local data into training and validation sets with an 80:20 ratio. The training set is used for semantic feature extraction and model updates on the vehicle-side, while the validation set is used for assessing local model performance and tuning hyperparameters.

Table 1: Parameter settings for different network condition levels

Network Level	Bandwidth B (Mbps)	Initial Delay δ_0 (ms)	Packet Loss Rate P (%)
Excellent	50.0	10	0.01
Good	20.0	30	0.1
Fair	5.0	100	1
Poor	1.0	500	5

Communication Settings: As presented in Table 1, this paper defines four distinct network condition levels, which are randomly assigned to $V = 5$ vehicles in a cyclic manner. Vehicle initial positions are randomly distributed within a circular area centered at the base station, with a radius of $r = 1000$ meters. Vehicle speed S_{v_i} is uniformly distributed over $[0, 30]$ m/s. The distance coefficient is set to $\beta = 0.5$. The speed factor upper limit is set to $\varepsilon = 2.0$. The random perturbation factor follows $\eta \sim \mathcal{N}(1.0, 0.2^2)$. The maximum delay round is set to $S_{\max} = 2$, and the global delay scaling factor is set to $\omega = 1.0$. Under the weighting strategy, the weight parameter is set to $\gamma = 0.1$. The network reliability parameter is set to $R = 0.99$, the retry decay factor is set to $\tau^{(N)} = 0.9$, and the maximum number of retransmissions is set to $N = 3$. In the experiments, since the feature data from SeFL is continuously utilized and regarded as newly added samples, the "none" strategy is applied to SeFL with its weight consistently set to

1.0. For TFL and HFL, during the training of the classification task head, employing the "none" or "weight" strategies leads to gradient explosion issues. Consequently, only the "drop" strategy can be adopted for both TFL and HFL. As for FTFL, it employs the "weight" strategy to ensure full utilization of stale information.

Neural Network Structure: Pre-trained ResNet [43] models on the ImageNet dataset are used as semantic encoders. Before federated training begins, the base station distributes pre-trained encoders to all vehicles. Taking ResNet-101 as an example, the SeFL network structure, as illustrated in Figure 6, consists of the following components:

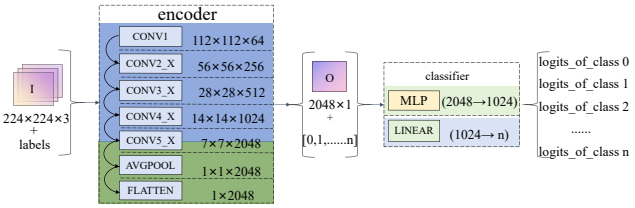


Figure 6: Composition of SeFL Network Structure

The encoder employs a pre-trained ResNet-101 NN model, consisting of 1 convolutional layer, 4 groups of residual blocks (CONV2_X, CONV3_X, CONV4_X, and CONV5_X containing 3, 4, 23, and 3 residual blocks respectively) [44], and a global average pooling layer, which ultimately outputs a 2048-dimensional semantic feature vector. The feature representations learned by the pre-trained model on ImageNet exhibit strong generalization capability, enabling the capture of semantic information ranging from low-level edges and textures to high-level abstractions. In SeFL training process, the encoder parameters remain frozen, avoiding large-scale parameter updates and reducing the computational burden on vehicle-side. To validate the impact of different model scales on SeFL performance, experiments are also conducted using ResNet-18, ResNet-34, and ResNet-50 NN models.

The classification task head adopts a single-layer fully connected network with an input dimension of 2048 and an output dimension of 43 (corresponding to the number of categories in the GTSRB dataset), utilizing Softmax as the activation function. For SeFL, this classification task head is trained exclusively on the server-side, while for TFL, FTFL, and HFL, the head participates in distributed training across vehicles and the server. Training employs the SGD optimizer with a learning rate of 0.01 and a momentum coefficient of 0.9. The loss function combines cross-entropy loss with L2 regularization, where the regularization coefficient is set to $\lambda = 10^{-4}$ for all methods.

Comparison Baseline Settings: To evaluate the effectiveness of the SeFL framework, this paper designs three groups of comparative experiments.

Traditional federated learning (TFL) requires the vehicle-side to maintain a complete deep learning model, including the feature extraction encoder and the classification task head, and upload the updated model parameters to the server. In TFL, the encoder is trainable, and vehicles perform 5 local epochs of training per round using their local datasets before uploading the complete model parameters, which include both encoder and classifier, to the server for aggregation.

Fine-tuning FL (FTFL) adopts a strategy similar to SeFL, using a pre-trained ResNet encoder for feature extraction, keeping the encoder frozen, and performing federated training on the task head. Although the encoder remains frozen like SeFL, the key difference is that the classification task head is trained locally on each vehicle for 5 local epochs, and only the classifier parameters are uploaded to the server for aggregation. This requires vehicles to perform backpropagation for the classifier, unlike SeFL's server-only training approach. After feature extraction and task head training, vehicle sides upload the parameters to the server for aggregation.

Hierarchical FL (HFL) [26] divides the NN between vehicle sides and the server for hierarchical training. The ResNet encoder is divided into two parts, i.e., the vehicle-side encoder E_1, which uses pre-trained parameters and is kept frozen for preliminary feature extraction, and the server-side encoder E_2, which participates in training for deep semantic extraction. The task head is split into the server-side task head C_1 and the vehicle-side task head C_2, both participating in separate training.

Taking ResNet-101 as an example, Figure 7 depicts the network structure of HFL. The training process consists of four stages. Specifically, first, the vehicle-side uses the frozen encoder E_1 to extract features from data x_i, producing blurred features $f_{\text{blur}} = E_1(x_i)$ (i.e., intermediate feature maps output by the shallow vehicle-side encoder E_1, which have not yet undergone deep semantic extraction by E_2). E_1 includes the convolutional layer, residual block groups 2-4 (CONV2_X to CONV4_X), and the first residual block of CONV5_X (conv5_1), with the remaining residual block (conv5_2) allocated to the server. Second, the server uses E_2 for deep feature extraction, yielding $y_{\text{complete}} = E_2(f_{\text{blur}})$, and generates an intermediate representation $z_{\text{transform}} = C_1(y_{\text{complete}})$ using C_1. Third, the vehicle-side receives $z_{\text{transform}}$ and updates the parameters of C_2 ($\theta_{\mathbf{C_2}}$) using local labels. Finally, the server aggregates $\theta_{\mathbf{C_2}}$ and feature gradients from all vehicle sides to update E_2 and C_1. This hierarchical design enables collaborative training between the vehicle-sides and the server in HFL.

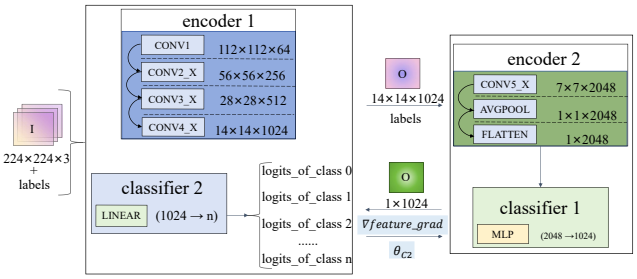


Figure 7: Composition of HFL Network Structure

Shared Training Settings: To ensure fair comparison, all methods share the following configurations: (1) Total training rounds: $R = 20$. (2) Optimizer: SGD with momentum coefficient $\mu = 0.9$. (3) Learning rate: $\eta = 0.01$ (fixed throughout training). (4) L2 regularization: $\lambda = 10^{-4}$. (5) Image preprocessing: resize to 224×224 pixels with ImageNet normalization ($\mu = [0.485, 0.456, 0.406]$, $\sigma = [0.229, 0.224, 0.225]$).

Method-Specific Differences: The key differences among methods are: (1) Encoder training status: SeFL, FTFL, and HFL-E_1 keep encoders frozen; TFL and HFL-E_2 have trainable encoders. (2) Training location: SeFL trains the classifier exclusively on the server; TFL, FTFL, and HFL train classifiers on vehicles. (3) Local epochs: SeFL has no local epochs; TFL, FTFL, and HFL perform 5 local epochs per round. (4) Upload content: SeFL uploads semantic features; TFL uploads complete model parameters; FTFL uploads classifier parameters; HFL uploads blurred features and the parameters of C_2.

Evaluation Metrics: To evaluate the performance of the SeFL framework, this paper assesses it based on three metrics, i.e., model effectiveness (classification accuracy), interaction overhead, and convergence speed. Model effectiveness reflects the success of SeFL's server-side centralized training in mitigating the impact of non-IID data. Additionally, by setting different $T_{\text{threshold}}$ to simulate various network delay conditions, the performance stability of each method can be validated. Interaction overhead quantifies the communication efficiency achieved by replacing model parameter transmission with semantic features. Convergence speed reflects the optimization of system iteration efficiency achieved during training.

This paper quantifies interaction overhead from the perspective of communication volume. Per-round uplink communication volume quantifies the average data transmitted from vehicle-side to the server during each training round, calculated as $C_{\text{up}}^{\text{round}} = \frac{1}{N}\sum_{i=1}^{N} |M_i^{\text{upload}}|$, where N denotes the number of vehicles participating in the current training round, and $|M_i^{\text{upload}}|$ represents the number of bytes uploaded by vehicle-side i. All numerical values are encoded in FP32 (32-bit floating-point, 4 bytes per value) for model parameters and features, and in int32 (4 bytes) for labels. For TFL and FTFL, the uplink data consist of model parameters. For SeFL, the uplink data consist of semantic feature vectors and corresponding labels. For HFL, the uplink data consist of blurred features, corresponding labels, the parameters of the classification task head C_2, and feature gradients. Per-round downlink communication volume quantifies the average data transmitted from the server to vehicle-side during each training round, calculated as $C_{\text{down}}^{\text{round}} = \frac{1}{N}\sum_{i=1}^{N} |M_i^{\text{download}}|$, where $|M_i^{\text{download}}|$ represents the number of bytes distributed from the server to vehicle-side i. Specifically, for SeFL and FTFL, the downlink transmission includes only the parameters of the classification task head C_2; for TFL, it includes all model parameters; and for HFL, it includes the global model parameters along with intermediate computational results. Total communication volume represents the cumulative communication overhead over all training rounds, i.e., the sum of per-round uplink and downlink communication volumes.

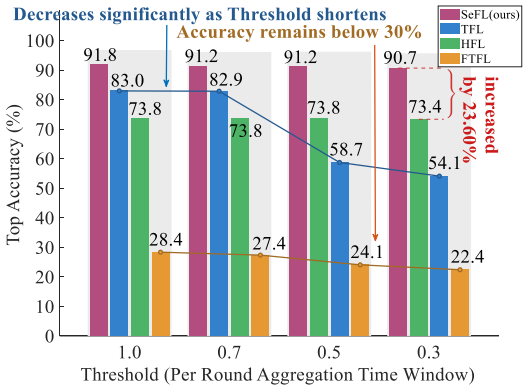
To quantify the training efficiency of the model, convergence speed is adopted as the evaluation metric, which is defined as the average time per round required for the model to reach convergence, measured in seconds. This metric comprehensively reflects both local computation time and communication transmission time.

4.2 Effectiveness Experiments of the Proposed SeFL

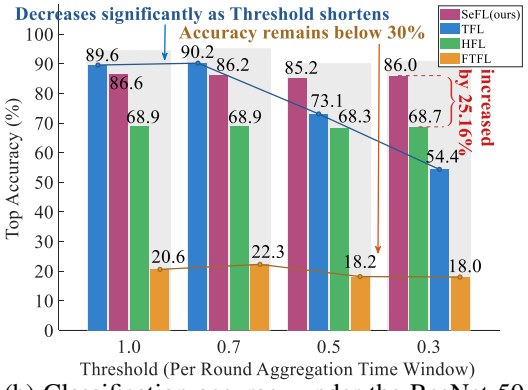
To validate the effectiveness of the proposed SeFL, this subsection compares the classification performance of SeFL with three baseline methods (TFL, HFL, and FTFL) on the GTSRB dataset under different $T_{\text{threshold}}$ values and encoder configurations. All methods use the same pre-trained ResNet series encoders, and classification performance is measured by accuracy. The experiments set four $T_{\text{threshold}}$ values of 1.0 s, 0.7 s, 0.5 s, and 0.3 s to simulate various communication delay conditions. Figure 8(a–d) shows the classification accuracy under four encoder configurations, namely ResNet-101, ResNet-50, ResNet-34, and ResNet-18, where the horizontal axis represents the $T_{\text{threshold}}$ and the vertical axis represents the classification accuracy.

As shown in Figure 8(a), under the configuration of the ResNet-101 NN model with a $T_{\text{threshold}}$ of 1.0 s, the proposed SeFL method achieves an accuracy of 91.8%, representing an improvement of 8.8% over TFL, 18% over HFL, and 63.4% over FTFL. This advantage is attributed to its architectural design, where the server centrally receives and aggregates the semantic features uploaded from all vehicle sides, enabling the classification task head to be optimized based on a training set that reflects the global data distribution. This effectively mitigates the negative impact of non-IID data on model performance. Notably, the semantic extraction accuracy of the FTFL method does not exceed 30%, which is far below the performance required for practical applications. The reason is that, although FTFL employs a pre-trained encoder for feature extraction, the encoder parameters remain frozen, and only the classification task head is trained locally on each vehicle. Afterward, the task head parameters are uploaded to the server for federated aggregation. Due to the non-IID distribution of data across vehicle sides, the locally trained task head parameters are biased, and simple parameter aggregation cannot effectively integrate heterogeneous feature spaces, leading to significant degradation in the global model's performance.

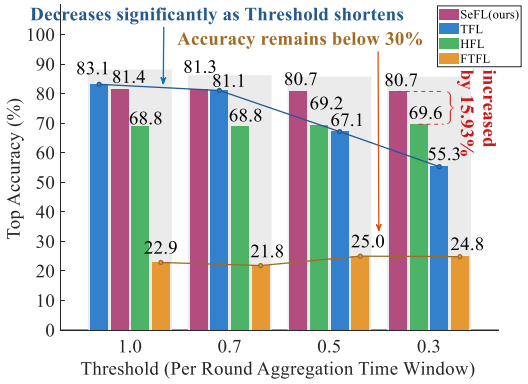
Figure 8(a) shows that the accuracy of TFL decreases significantly as $T_{\text{threshold}}$ shortens. When the threshold drops to 0.3 seconds, the accuracy plummets by about 28.9%. The fundamental cause of this phenomenon lies in the incompatibility between delayed parameter versions and the current round, leading to gradient conflicts and deviations in the optimization direction, ultimately resulting in model performance degradation. In contrast, SeFL's accuracy remains almost unaffected by changes in the $T_{\text{threshold}}$, demonstrating excellent stability. This stability arises from SeFL's semantic feature-based training paradigm, which treats delayed feature data as additional training samples in model iteration, thus avoiding the convergence fluctuations caused by outdated parameter aggregation in TFL and enhancing system stability under communication delay conditions. Although HFL and FTFL also show stability, their accuracy is significantly lower than SeFL. This is because HFL transmits a small amount of model parameters during training, and in the experiments, a discard strategy is applied to the delayed model parameters. While this strategy prevents drastic accuracy fluctuations, it sacrifices model precision, leading to a gap between HFL's



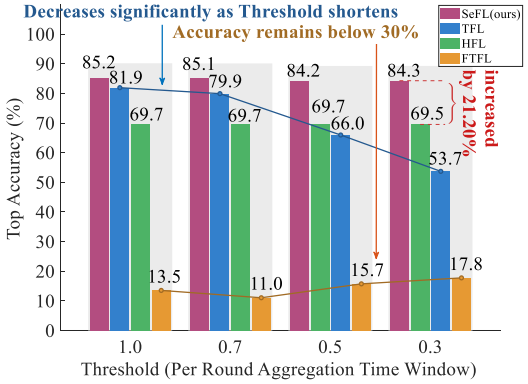
(a) Classification accuracy under the ResNet-101 configuration



(b) Classification accuracy under the ResNet-50 configuration



(c) Classification accuracy under the ResNet-34 configuration



(d) Classification accuracy under the ResNet-18 configuration

Figure 8: Classification accuracy under different ResNet configurations

final accuracy and that of SeFL. Even under the 0.3 s $T_{\text{threshold}}$ condition, SeFL still achieves an accuracy that is 23.6% higher than that of TFL.

Horizontally analyzing the experimental results in Figure 8(b–d), under the ResNet-50, ResNet-34, and ResNet-18 NN configurations, SeFL maintains a leading advantage in overall accuracy. Although under the ResNet-50 and ResNet-34 models with a $T_{\text{threshold}}$ of 1.0 s, SeFL's accuracy is slightly lower than that of TFL, the gap is kept within 3%. As the $T_{\text{threshold}}$ decreases, TFL's accuracy drops significantly, while SeFL remains consistently stable.

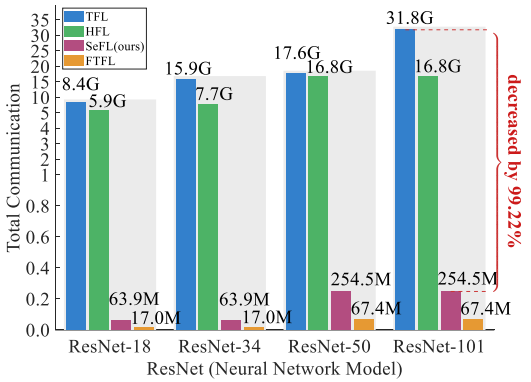
In summary, SeFL outperforms the baseline methods in both classification accuracy and stability. These conclusions hold across different encoder configurations, indicating that the stability and accuracy of the SeFL framework are not restricted to specific NN models, demonstrating strong adaptability in complex environments.

4.3 Communication Overhead Evaluation

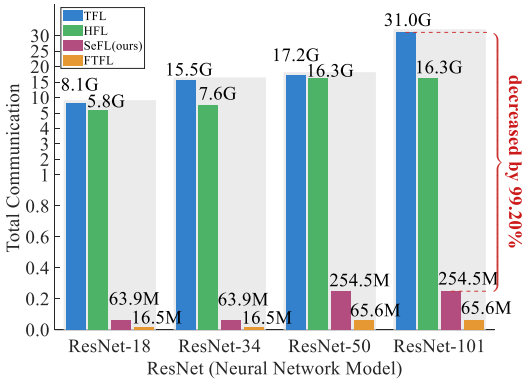
To validate the advantages of SeFL in reducing communication overhead, this subsection compares the total communication volume of four methods under different $T_{\text{threshold}}$ values and encoder configurations. Communication volume, as a key metric for measuring interaction overhead, directly reflects the extent of network bandwidth resource utilization by the system. Figure 9(a–d) shows the total communication volume under $T_{\text{threshold}}$ configurations of 1.0 s, 0.7 s, 0.5 s, and 0.3 s, where the horizontal axis represents the encoder type (ResNet-18/34/50/101) and the vertical axis represents the total communication volume.

As shown in Figure 9(a), it can be observed that under the $T_{\text{threshold}}$ configuration of 1.0 s, the communication volume of all four methods increases with the encoder depth. Under the ResNet-101 model, TFL has the highest total communication volume, reaching 31.8 GB, followed by HFL with 16.8 GB, while SeFL's communication volume is 254.5 MB, representing a reduction of 99.2% compared to TFL and 98.5% compared to HFL. Since TFL requires uploading and downloading complete model parameters between vehicle-side and the server, it has the highest communication volume. Although HFL transmits partial semantic information, it uploads blurred features output by the shallow encoder with insufficient compression, and also requires downloading deep encoder parameters, leading to relatively high communication overhead. In contrast, SeFL only uploads compressed semantic features and downloads the single-layer classification task head parameters, resulting in lower communication volume. FTFL has the lowest communication volume at 67.4 MB, because FTFL only uploads or downloads classification task head parameters, and the task head is a single fully connected layer. Although FTFL has the lowest communication volume, as shown in the experimental results from Section 4.2, its classification accuracy consistently remains below 30%, failing to meet practical application requirements. Therefore, low communication volume cannot compensate for its performance degradation.

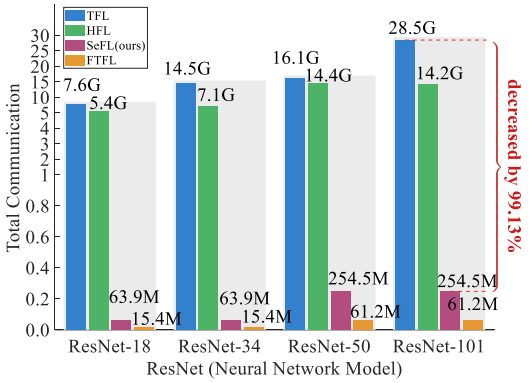
We find that under the ResNet-18 neural network configuration, the communication overhead values are 8.4 GB, 5.9 GB, and 63.9 MB for TFL, HFL, and SeFL, respectively.



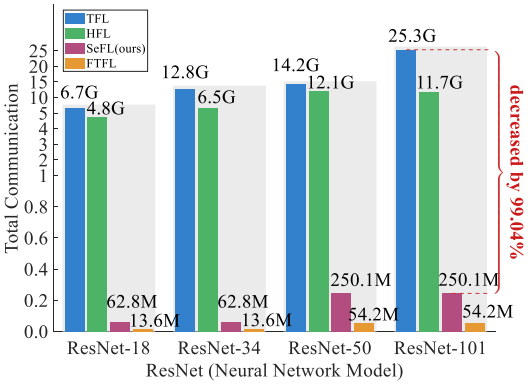
(a) Total traffic under the $T_{\text{threshold}}$ = 1.0 s configuration



(b) Total traffic under the $T_{\text{threshold}}$ = 0.7 s configuration



(c) Total traffic under the $T_{\text{threshold}}$ = 0.5 s configuration



(d) Total traffic under the $T_{\text{threshold}}$ = 0.3 s configuration

Figure 9: Total communication traffic under different threshold configurations

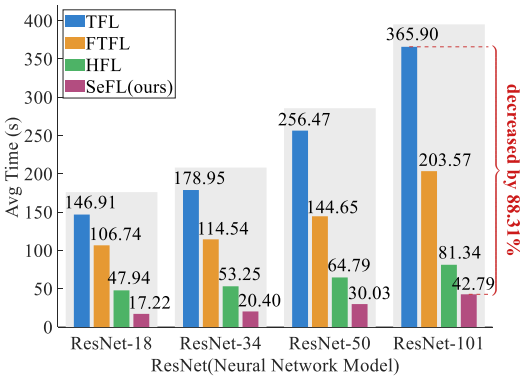
The communication overhead of TFL remains two orders of magnitude higher than that of SeFL. This indicates that even under shallower network structures, SeFL's communication efficiency advantage remains significant. Analyzing the experimental results in Figure 9(b–d) under $T_{\text{threshold}}$ configurations of 0.7 s, 0.5 s, and 0.3 s, SeFL's communication volume advantage over TFL and HFL remains consistent across different conditions.

In summary, SeFL outperforms the TFL and HFL baseline methods in terms of communication overhead. Although FTFL has a lower communication volume than SeFL, its accuracy is too low. Considering both accuracy and communication overhead, SeFL achieves optimal communication efficiency while maintaining high accuracy. Furthermore, these conclusions hold under different encoder configurations and $T_{\text{threshold}}$ conditions, validating the generalizability of the SeFL method. Therefore, SeFL is better suited for vehicular network environments with limited network bandwidth resources, particularly for practical application scenarios such as large-scale vehicle collaborative training and edge computing with limited resources.

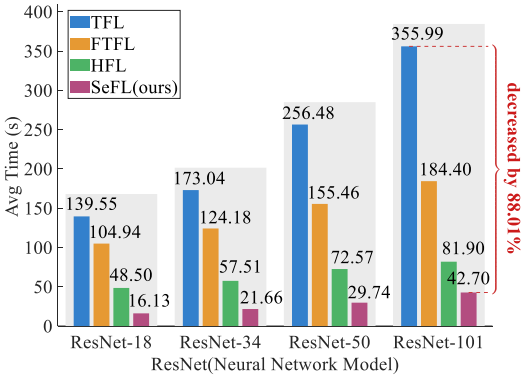
4.4 Convergence Speed Evaluation

To validate the performance of SeFL in improving training efficiency, this subsection compares the convergence speed of four methods under different $T_{\text{threshold}}$ values and encoder configurations. Convergence speed is measured by the average per-round iteration time, which comprehensively reflects both local computation time and communication transmission time. Figure 10 shows the average per-round iteration time under $T_{\text{threshold}}$ configurations of 1.0 s, 0.7 s, 0.5 s, and 0.3 s, where the horizontal axis represents the encoder type (ResNet-18/34/50/101) and the vertical axis represents the average per-round iteration time.

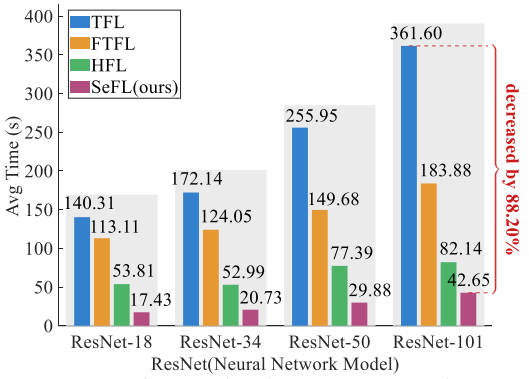
As shown in Figure 10(a), it can be observed that under the $T_{\text{threshold}}$ configuration of 1.0 s, the average per-round iteration time of all four methods increases as the encoder depth increases. Under the ResNet-101 model, TFL has the longest per-round time, reaching 365.90 s. TFL has the longest per-round time because it requires vehicle sides to complete multiple iterations of local training and update model parameters, which increases local computation time, while the transmission of complete model parameters further extends the per-round time. FTFL has the second longest per-round time at 203.57 s, because FTFL also requires waiting for multiple vehicle-sides to complete local training and upload partial model parameters, and vehicle sides still need to perform model update training, increasing training time. HFL has a per-round time of 81.34 s, as HFL allows vehicle sides to perform only partial model training through model splitting, reducing local computation time and transmission data volume compared to TFL and FTFL. SeFL has the shortest average per-round time at only 42.79 s, achieving an 88.31% reduction in per-round iteration time compared to TFL. This is because SeFL offloads the training process to the server, eliminating the need for backpropagation updates on vehicle sides and only transmitting compressed semantic features, resulting in minimal local computation time, thus achieving the shortest per-round iteration time and the fastest convergence



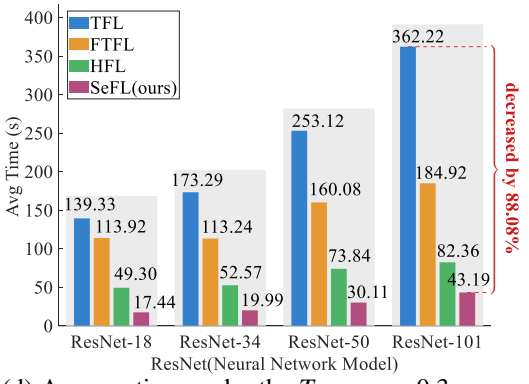
(a) Average time under the $T_{\text{threshold}}$ = 1.0 s configuration



(b) Average time under the $T_{\text{threshold}}$ = 0.7 s configuration



(c) Average time under the $T_{\text{threshold}}$ = 0.5 s configuration



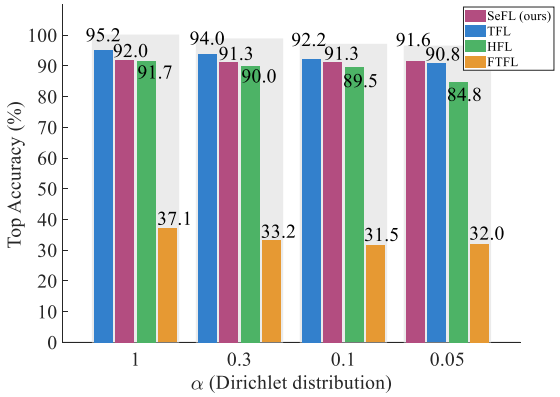
(d) Average time under the $T_{\text{threshold}}$ = 0.3 s configuration

Figure 10: Average time under different threshold configurations

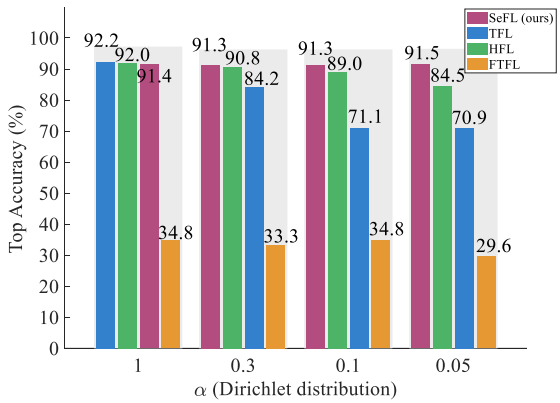
speed. per-round iteration time and the fastest convergence speed. Analyzing the experimental results in Figure 10(b–d) under $T_{\text{threshold}}$ configurations of 0.7 s, 0.5 s, and 0.3 s, it can be observed that SeFL's advantage in per-round iteration time over the other three methods remains consistent across different $T_{\text{threshold}}$. In summary, SeFL outperforms the other three baseline methods in terms of convergence speed. Furthermore, these conclusions hold under different encoder configurations and $T_{\text{threshold}}$ conditions, validating the generalizability of the SeFL method. Therefore, SeFL achieves faster convergence, making it more suitable for vehicular network scenarios that require high model update timeliness.

4.5 Non-IID Robustness Evaluation

To further validate the consistency of SeFL's advantage in handling different degrees of non-IID data, we conducted experiments using Dirichlet distributions with varying concentration parameters $\alpha \in \{1, 0.3, 0.1, 0.05\}$ to control the degree of data non-IID among vehicles. The parameter α determines the data distribution pattern. Specifically, $\alpha = 1$ generates moderate non-IID, while a smaller value such as $\alpha = 0.05$ leads to stronger non-IID. Figure 11 presents the results comparing SeFL with baseline methods based on ResNet-101 under different α values. Two representative time thresholds of $T_{\text{threshold}} = 1.0$ s and $T_{\text{threshold}} = 0.3$ s are selected for evaluation.



(a) Classification accuracy under the $T_{\text{threshold}} = 1.0$ s configuration



(b) Classification accuracy under the $T_{\text{threshold}} = 0.3$ s configuration

Figure 11: Classification accuracy under different degrees of non-IID

As shown in Figure 11(a), when $T_{\text{threshold}} = 1.0$ s and $\alpha = 1$, TFL achieves an accuracy of approximately 95.6%, slightly outperforming SeFL, which achieves 91.8%. However, when α drops to 0.05, TFL's accuracy plummets to 71.3%, while SeFL maintains stable performance above 90%. Figure 11(b) reveals that when $T_{\text{threshold}} = 0.3$ s, due to the combined effects of parameter aggregation delay and non-IID data, TFL's performance declines more severely across all α values, whereas SeFL consistently maintains an accuracy above 91%.

The results indicate that as the degree of data non-IID increases, the advantage of SeFL over the baseline further expands. This clearly demonstrates SeFL's strong ability to handle non-IID challenges, which are key bottlenecks in practical IoV-FL systems.

5 Conclusion

This paper proposed a semantic FL framework for IoV, termed SeFL, to address the challenges of high communication overhead, convergence degradation caused by communication latency, and performance degradation due to non-IID data in traditional FL. SeFL divided the model into a pre-trained encoder on the vehicle-side and a classification task head on the server. The vehicle-side only uploaded compressed semantic features instead of model parameters, significantly reducing communication overhead. The server aggregated multi-source semantic features to reconstruct the global data distribution, thereby mitigating the impact of non-IID data. Meanwhile, delayed features are treated as new samples to avoid convergence fluctuations caused by outdated parameter aggregation. Under the ResNet-101 configuration with $T_{\text{threshold}}$ set to 1.0 s, experimental evaluation reveals that SeFL attains 91.8% classification accuracy, exceeding TFL and HFL by 8.8% and 18.0%, respectively. Furthermore, SeFL demonstrates substantial improvements in communication efficiency and convergence performance, achieving 99.2% and 98.5% reductions in communication overhead compared to TFL and HFL, along with 64.9% and 21.1% faster convergence, respectively. When $T_{\text{threshold}}$ is shortened to 0.3 s, SeFL exhibits stable accuracy while TFL suffers an approximately 28.9% degradation.

Funding

National Natural Science Foundation of China (NSFC) (62571028 and 62071033); Changping Innovation Joint Fund of Beijing Natural Science Foundation (L234084).

Author Contribution

The manuscript was written with contributions from all authors. Conceptualization, Man Luo, Jin Mao, Ge Xiong and Muhammad Rizwan Anjum; methodology, Man Luo, Jin Mao, Ge Xiong and Muhammad Rizwan Anjum; software, Man Luo; validation, Man Luo and Jin Mao; formal analysis, Man Luo and Jin Mao; investigation, Man Luo; resources, Ge Xiong; data curation, Man Luo; writing-original draft preparation, Man Luo and Jin Mao; writing-review and editing, Man Luo, Jin Mao, Ge Xiong and Muhammad Rizwan Anjum; visualization, Man Luo; supervision, Ge Xiong and Jin Mao; project administration, Ge Xiong; funding acquisition, Ge Xiong. All authors have read and agreed to the published version of the manuscript.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability

The GTSRB dataset is publicly available at https://benchmark.ini.rub.de/gtsrb_dataset.html.

References

[1] Contreras-Castillo, J., Zeadally, S., Guerrero-Ibañez, J.A.: Internet of vehicles: architecture, protocols, and security. IEEE Internet of Things Journal **5**(5), 3701–3709 (2018). https://doi.org/10.1109/JIOT.2017.2690902

[2] Fadhil, J.A., Sarhan, Q.I.: Internet of vehicles (IoV): a survey of challenges and solutions. In Proceedings of the 21st International Arab Conference on Information Technology (ACIT), pp. 1–10 (2020). https://doi.org/10.1109/ACIT50332.2020.9300095

[3] Kiran, B.R., Sobh, I., Talpaert, V., Mannion, P., Sallab, A.A.A., Yogamani, S., Perez, P.: Deep reinforcement learning for autonomous driving: a survey. IEEE Transactions on Intelligent Transportation Systems **23**(6), 4909–4926 (2022). https://doi.org/10.1109/TITS.2021.3054625

[4] Kuutti, S., Bowden, R., Jin, Y., Barber, P., Fallah, S.: A survey of deep learning applications to autonomous vehicle control. IEEE Transactions on Intelligent Transportation Systems **22**(2), 712–733 (2021). https://doi.org/10.1109/TITS.2019.2962338

[5] Grigorescu, S., Trasnea, B., Cocias, T., Macesanu, G.: A survey of deep learning techniques for autonomous driving. Journal of Field Robotics **37**(3), 362–386 (2020). https://doi.org/10.1002/rob.21918

[6] Huang, Y., Du, J., Yang, Z., Zhou, Z., Zhang, L., Chen, H.: A survey on trajectory-prediction methods for autonomous driving. IEEE Transactions on Intelligent Vehicles **7**(3), 652–674 (2022). https://doi.org/10.1109/TIV.2022.3167103

[7] Zhang, P., Wang, G., Chen, S., Yu, Y., Chen, L.: The application of mobile edge computing in the space-air-ground integrated network. Journal of Intelligent Computing and Networking **1**(1), 71–97 (2025). https://doi.org/10.64509/jicn.11.13

[8] Abboud, K., Omar, H.A., Zhuang, W.: Interworking of

DSRC and cellular network technologies for V2X communications: a survey. IEEE Transactions on Vehicular Technology **65**(12), 9457–9470 (2016). https://doi.org/10.1109/TVT.2016.2591558

[9] Moradi-Pari, E., Tian, D., Bahramgiri, M., Rajab, S., Bai, S.: DSRC versus LTE-V2X: empirical performance analysis of direct vehicular communication technologies. IEEE Transactions on Intelligent Transportation Systems **24**(5), 4889–4903 (2023). https://doi.org/10.1109/TITS.2023.3247339

[10] Zadobrischi, E., Havriliuc, S.: Enhancing scalability of C-V2X and DSRC vehicular communication protocols with LoRa 2.4 GHz in the scenario of urban traffic systems. Electronics **13**(14), 2845 (2024). https://doi.org/10.3390/electronics13142845

[11] Yang, Z., Cheng, C., Li, Z., Wang, R., Zhang, X.: Reliable federated learning based on delayed gradient aggregation for intelligent connected vehicles. Engineering Applications of Artificial Intelligence **140**, 109719 (2025). https://doi.org/10.1016/j.engappai.2024.109719

[12] Chellapandi, V.P., Yuan, L., Brinton, C.G., Zak, S.H., Wang, Z.: Federated learning for connected and automated vehicles: a survey of existing approaches and challenges. IEEE Transactions on Intelligent Vehicles **9**(1), 119–137 (2024). https://doi.org/10.1109/TIV.2023.3332675

[13] Almanifi, O.R.A., Chow, C.-O., Tham, M.-L., Chuah, J.H., Kanesan, J.: Communication and computation efficiency in federated learning: a survey. Internet of Things **22**, 100742 (2023). https://doi.org/10.1016/j.iot.2023.100742

[14] Chen, M., Poor, H.V., Saad, W., Cui, S.: Convergence time optimization for federated learning over wireless networks. IEEE Transactions on Wireless Communications **20**(4), 2457–2471 (2021). https://doi.org/10.1109/TWC.2020.3042530

[15] Dinh, C.T., Tran, N.H., Nguyen, M.N.H., Hong, C.S., Bao, W., Zomaya, A.Y., Gramoli, V.: Federated learning over wireless networks: convergence analysis and resource allocation. IEEE/ACM Transactions on Networking **29**(1), 398–409 (2021). https://doi.org/10.1109/TNET.2020.3035770

[16] Zhu, L., Lin, H., Lu, Y., Lin, Y., Han, S.: Delayed gradient averaging: tolerate the communication latency for federated learning. In Proceedings of the 35th International Conference on Neural Information Processing System, pp. 29995–30007 (2021)

[17] Chen, M., Mao, B., Ma, T.: FedSA: a staleness-aware asynchronous federated learning algorithm with non-IID data. Future Generation Computer Systems **120**, 1–12 (2021). https://doi.org/10.1016/j.future.2021.02.012

[18] Qiang, X., Chang, Z., Ye, C., Hamalainen, T., Min, G.: Split federated learning empowered vehicular edge intelligence: concept, adaptive design, and future directions. IEEE Wireless Communications **32**(4), 90–97 (2025). https://doi.org/10.1109/MWC.009.2400219

[19] Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-IID data. arXiv preprint arXiv:1806.00582 (2018). https://doi.org/10.48550/arXiv.1806.00582

[20] Mao, J., Xiong, K., Liu, M., Qin, Z., Chen, W., Fan, P., Letaief, K.B.: A GAN-based semantic communication for text without CSI. IEEE Transactions on Wireless Communications **23**(10), 14498–14514 (2024). https://doi.org/10.1109/TWC.2024.3415363

[21] Xie, H., Qin, Z., Li, G.Y., Juang, B.-H.: Deep learning enabled semantic communication systems. IEEE Transactions on Signal Processing **69**, 2663–2675 (2021). https://doi.org/10.1109/TSP.2021.3071210

[22] Zhang, R., Xiong, K., Du, H., Niyato, D., Kang, J., Shen, X., Poor, H.V.: Generative AI-enabled vehicular networks: fundamentals, framework, and case study. IEEE Network **38**(4), 259–267 (2024). https://doi.org/10.1109/MNET.2024.3391767

[23] Zheng, G., Ni, Q., Navaie, K., Pervaiz, H., Zarakovitis, C.: A distributed learning architecture for semantic communication in autonomous driving networks for task offloading. IEEE Communications Magazine **61**(11), 64–68 (2023). https://doi.org/10.1109/MCOM.002.2200765

[24] Liu, J., Lu, Y., Wu, H., Dai, Y.: Efficient resource allocation and semantic extraction for federated learning empowered vehicular semantic communication. In Proceedings of the IEEE 98th Vehicular Technology Conference (VTC-Fall), pp. 1–5 (2023). https://doi.org/10.1109/VTC2023-Fall60731.2023.10333738

[25] Xiong, K., Zhang, Y., Fan, P., Yang, H.-C., Zhou, X.: Mobile service amount based link scheduling for high-mobility cooperative vehicular networks. IEEE Transactions on Vehicular Technology **66**(10), 9521–9533 (2017). https://doi.org/10.1109/TVT.2017.2714863

[26] Zheng, G., Ni, Q., Navaie, K., Pervaiz, H., Min, G., Kaushik, A., Zarakovitis, C.: Mobility-aware split-federated learning with transfer learning for vehicular semantic communication networks. IEEE Internet of Things Journal **11**(10), 17237–17248 (2024). https://doi.org/10.1109/JIOT.2024.3360230

[27] Sang, Z., Meng, C., Yan, W., Gao, B., Xiong, K., Fan, P.: Joint aggregation node selection and power adjustment for federated learning in IoV. In Proceedings of the 5th International Conference on Computer, Control and Robotics (ICCCR), pp. 438–443 (2025). https://doi.org/10.1109/ICCCR65461.2025.11072661

[28] Paragliola, G.: Evaluation of the trade-off between performance and communication costs in federated learning scenario. Future Generation Computer Systems **136**, 282–293 (2022). https://doi.org/10.1016/j.future.2022.06.006

[29] McMahan, B., Moore, E., Ramage, D., Hampson, S., Arcas, B.A.y.: Communication-efficient learning of deep networks from decentralized data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS), pp. 1273–1282 (2017)

[30] Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision (ECCV), pp. 818–833 (2014). https://doi.org/10.1007/978-3-319-10590-1_53

[31] Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 512–519 (2014). https://doi.org/10.1109/CVPRW.2014.131

[32] Yosinski, J., Clune, J., Bengio, Y., Lipson, H.: How transferable are features in deep neural networks? In Advances in Neural Information Processing Systems 27, pp. 1–9 (2014)

[33] Lu, F., Zhu, K., Zhai, W., Cao, Y., Zha, Z.-J.: Likelihood-aware semantic alignment for full-spectrum out-of-distribution detection. Journal of Intelligent Computing and Networking **1**(1), 1–13 (2025). https://doi.org/10.64509/jicn.11.10

[34] Molisch, A.F., Tufvesson, F., Karedal, J., Mecklenbrauker, C.F.: A survey on vehicle-to-vehicle propagation channels. IEEE Wireless Communications **16**(6), 12–22 (2009). https://doi.org/10.1109/MWC.2009.5361174

[35] Wu, D., Negi, R.: Effective capacity: a wireless link model for support of quality of service. IEEE Transactions on Wireless Communications **2**(4), 630–643 (2003). https://doi.org/10.1109/TWC.2003.814353

[36] 3GPP: Study on channel model for frequencies from 0.5 to 100 GHz. Technical Report TR 38.901 (Release 16), 3rd Generation Partnership Project (2020)

[37] Boban, M., Vinhoza, T.T.V., Ferreira, M., Barros, J., Tonguz, O.K.: Impact of vehicles as obstacles in vehicular ad hoc networks. IEEE Journal on Selected Areas in Communications **29**(1), 15–28 (2011). https://doi.org/10.1109/JSAC.2011.110103

[38] Ho, Q., Cipar, J., Cui, H., Lee, S., Kim, J.K., Gibbons, P.B., Gibson, G.A., Xing, E.P.: More effective distributed ML via a stale synchronous parallel parameter server. In Proceedings of the 27th International Conference on Neural Information Processing Systems, pp. 1223–1231 (2013)

[39] Karagiannis, G., Altintas, O., Ekici, E., Heijenk, G., Jarupan, B., Lin, K., Weil, T.: Vehicular networking: a survey and tutorial on requirements, architectures, challenges, standards and solutions. IEEE Communications Surveys & Tutorials **13**(4), 584–616 (2011). https://doi.org/10.1109/SURV.2011.061411.00019

[40] Xiang, W., Gozalvez, J., Niu, Z., Altintas, O., Ekici, E.: Wireless access in vehicular environments. Journal on Wireless Communications and Networking **2009**, 576217 (2009). https://doi.org/10.1155/2009/576217

[41] Slomson, A.B.: Introduction to combinatorics. Chapman and Hall/CRC Press, Boca Raton, FL, USA (1997)

[42] Stallkamp, J., Schlipsing, M., Salmen, J., Igel, C.: The German Traffic Sign Recognition Benchmark: a multi-class classification competition. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), pp. 1453–1460 (2011). https://doi.org/10.1109/IJCNN.2011.6033395

[43] TorchVision: PyTorch's computer vision library. https://github.com/pytorch/vision Accessed: 2025-09-17

[44] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90